Name: Muhammad Ahad Imran

ID: F2360734

PFAI LAB MIDS

Q2)

1)

```
library(dplyr)

data(mtcars)


result <- mtcars %>%
  filter(mpg > 20) %>%
  arrange(desc(hp))


print(result)
```

OUTPUT

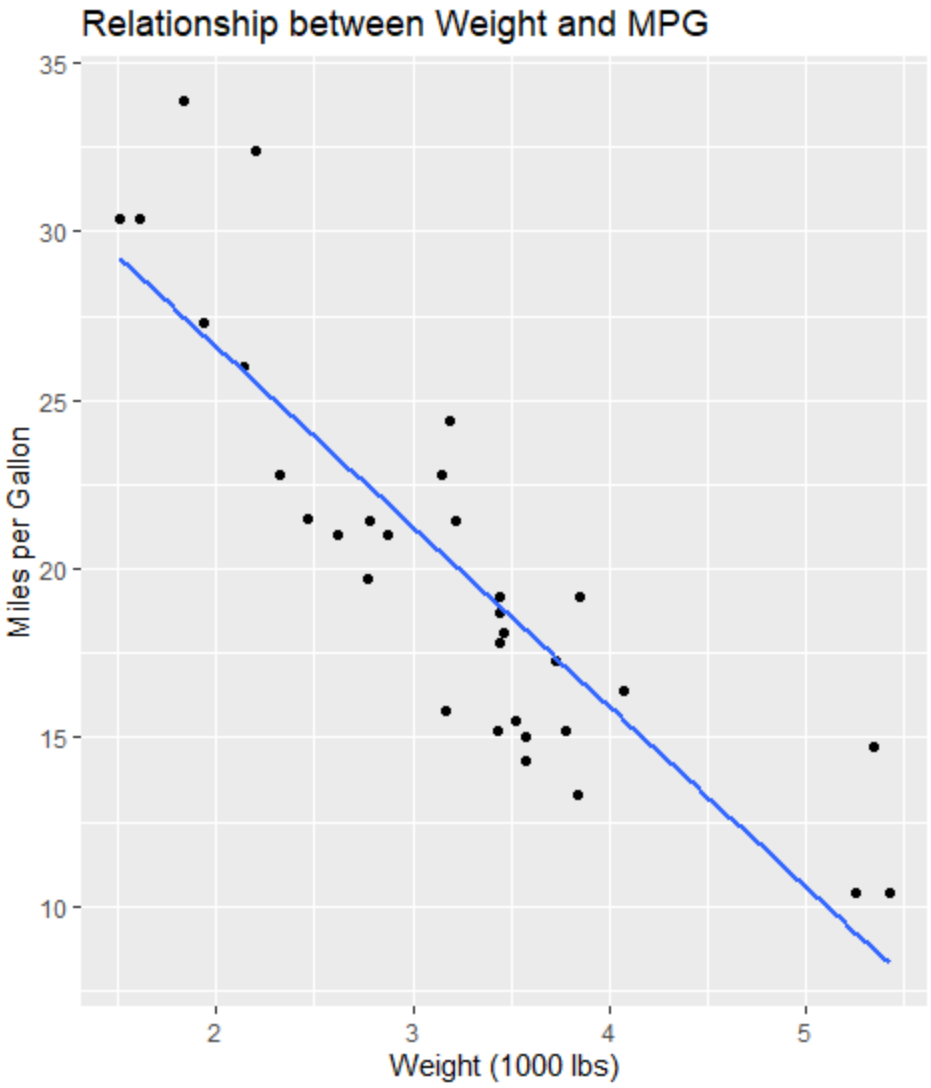
```
> print(result)
                mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
Toyota Corona  21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Merc 230       22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Fiat X1-9      27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Merc 240D      24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
```

2)

```
library(ggplot2)
data(mtcars)


ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Relationship between Weight and MPG",
       x = "Weight (1000 lbs)",
       y = "Miles per Gallon")
```

OUTPUT



3)

```
# Sample data

data <- c(9, 10, 11, 12, 8, 9, 10, 11)


# One-sample t-test

t.test(data, mu = 10)

print(result)
```

OUTPUT

```
        One Sample t-test

data:  data
t = 0, df = 7, p-value = 1
alternative hypothesis: true mean is not equal to 10
95 percent confidence interval:
  8.905392 11.094608
sample estimates:
mean of x
       10

> |
```

4)

```
library(caret)

data(mtcars)


# Split data into training and testing sets (80-20 split)

set.seed(123)

trainIndex <- createDataPartition(mtcars$mpg, p = 0.8, list = FALSE)

trainData <- mtcars[trainIndex, ]

testData <- mtcars[-trainIndex, ]


# Preprocess: scale the data

preProc <- preProcess(trainData, method = c("center", "scale"))

trainScaled <- predict(preProc, trainData)

testScaled <- predict(preProc, testData)


# Train linear regression model

model <- train(mpg ~ ., data = trainScaled, method = "lm")
```

```
# Predict on test set
predictions <- predict(model, newdata = testScaled)


# Evaluate with RMSE
rmse <- sqrt(mean((predictions - testScaled$mpg)^2))
print(paste("RMSE:", rmse))
```

OUTPUT

```
> # Evaluate with RMSE
> rmse <- sqrt(mean((predictions - testScaled$mpg)^2))
> print(paste("RMSE:", rmse))
[1] "RMSE: 0.84148582400261"
>
```